

Documentación de la Red Social con Node.js

Francisco J. Riedemann Wistuba - *UO252473@uniovi.es*

Manuel Junco Diez - *UO252010@uniovi.es*

*16/05/2018*

Universidad de Oviedo

Escuela de Ingeniería Informática

Sistemas Distribuidos e Internet

Grupo L03

# Casos de uso

## 1 Público: registrarse como usuario

***Los usuarios deben poder registrarse en la aplicación aportando un email, nombre y una contraseña (que deberá repetirse dos veces y coincidir entre sí). El email del usuario no podrá estar repetido en el sistema, se debe informar al usuario de los errores en el proceso de registro.***

Para el proceso de registro se dispone de signup.html el cual contiene un formulario donde se debe introducir email, nombre, password y la confirmación de la password. Este formulario contiene validaciones para no dejar en blanco los diferentes inputs, cuando los datos se han introducido, y se pulsa el botón de aceptar se envía una petición POST a /registrarse.

Esta petición es recibida por la función correspondiente en rusuarios.js. Aquí se realizar una serie de comprobaciones de los parámetros introducidos en el formulario, por ejemplo, la longitud de la password y el nombre, o que el email contenga un ‘@’. Luego se busca en la colección de ‘usuarios’ un usuario con el mismo email introducido, si no existe ninguno se encripta la contraseña, se introduce el nuevo usuario y se muestra un mensaje por pantalla con “nuevo usuario registrado”, por el contrario, si alguno de los puntos anteriores de comprobación fallara, se informa al usuario con una alerta.

## 2 Público: iniciar sesión

***Suministrando su email y contraseña, un usuario podrá autenticarse ante el sistema. Sólo los usuarios que proporcionen correctamente su email y su contraseña podrán iniciar sesión con éxito. En caso de que el inicio de sesión fracase, será necesario mostrar un mensaje de error indicando el problema. En caso de que el inicio de sesión sea correcto se debe dirigir al usuario a la vista que “lista todos los usuarios de la aplicación”.***

Para el proceso de identificación se dispone de login.html el cual contiene un formulario donde se debe introducir email y password. Este formulario envía los datos por medio de una petición POST a /identificarse.

Esta petición es recibida por la función correspondiente en rusuarios.js. Aquí se encripta la contraseña obtenida del formulario y se busca en la lista de usuarios un usuario con ese email y contraseña, en caso de que no se obtenga ningún usuario se envía un mensaje de alerta notificando el error, pero en caso de que sea correcto y se obtenga al usuario correspondiente se redirige a /usuario.

## 3 Usuario registrado: listar todos los usuarios de la aplicación

## 4 Usuario registrado: buscar entre todos los usuarios de la aplicación

## 5 Usuario registrado: enviar una invitación de amistad a un usuario

## 6 Usuario registrado: listar las invitaciones de amistad recibidas

## 7 Usuario registrado: aceptar una invitación recibida

## 8 Usuario registrado: listar los usuarios amigos

## S.1 Identificarse con usuario – token

## S.2 Usuario identificado: listar todos los amigos

## S.3 Usuario identificado: Crear un mensaje

## S.4 Usuario identificado: Obtener mis mensajes de una “conversación”

## \* S.5 Usuario identificado: Marcar mensaje como leído

## C.1. Autenticación del usuario

## C.2. Mostrar la lista de amigos

## C.3. Mostrar los mensajes

## C.4. Crear mensaje

# Pruebas automatizadas

## [RegVal]

## 1.2 [RegInval]

## 2.1 [InVal]

## 2.2 [InInVal]

## 3.1 [LisUsrVal]

## 3.2 [LisUsrInVal]

## 4.1 [BusUsrVal]

## 4.2 [BusUsrInVal]

## 5.1 [InvVal]

## 5.2 [InvInVal]

## 6.1 [LisInvVal]

## 7.1 [AcepInvVal]

## 8.1 [ListAmiVal]

## C1.1[CInVal]

## C1.2 [CInInVal]

## C.2.1 [CListAmiVal]

## C.2.2 [CListAmiFil]

## C3.1 [CListMenVal]

## C4.1 [CCrearMenVal]